



①9 BUNDESREPUBLIK
DEUTSCHLAND



DEUTSCHES
PATENTAMT

⑫ Übersetzung der
europäischen Patentschrift

⑧7 EP 0 426 410 B1

⑩ DE 690 27 510 T 2

⑤1 Int. Cl.⁶:
G 06 F 1/32

3)
DE 690 27 510 T 2

| | | |
|----|---|--------------|
| ②1 | Deutsches Aktenzeichen: | 690 27 510.2 |
| ⑧6 | Europäisches Aktenzeichen: | 90 311 832.1 |
| ⑧6 | Europäischer Anmeldetag: | 29. 10. 90 |
| ⑧7 | Erstveröffentlichung durch das EPA: | 8. 5. 91 |
| ⑧7 | Veröffentlichungstag der Patenterteilung beim EPA: | 19. 6. 96 |
| ④7 | Veröffentlichungstag im Patentblatt: | 23. 1. 97 |

③0 Unionspriorität: ③2 ③3 ③1
30.10.89 US 429270

⑦3 Patentinhaber:
Texas Instruments Inc., Dallas, Tex., US

⑦4 Vertreter:
Prinz und Kollegen, 81241 München

⑧4 Benannte Vertragsstaaten:
DE, FR, GB, IT, NL

⑦2 Erfinder:
Watts, Lavaughn F., Jr., Temple, Texas, US; Wallace,
Steven J., Temple, Texas, US

⑤4 Echtzeitleistungseinsparung für tragbare Rechner

Anmerkung: Innerhalb von neun Monaten nach der Bekanntmachung des Hinweises auf die Erteilung des europäischen Patents kann jedermann beim Europäischen Patentamt gegen das erteilte europäische Patent Einspruch einlegen. Der Einspruch ist schriftlich einzureichen und zu begründen. Er gilt erst als eingelegt, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

Die Übersetzung ist gemäß Artikel II § 3 Abs. 1 IntPatÜG 1991 vom Patentinhaber eingereicht worden. Sie wurde vom Deutschen Patentamt inhaltlich nicht geprüft.

DE 690 27 510 T 2

690 27 510.2-08

EP 0 426 410 (90311832.1-2215)

HINTERGRUND DER ERFINDUNG

Gebiet der Erfindung

Diese Erfindung bezieht sich auf Echtzeit-Leistungseinsparung bei Computern und insbesondere auf eine Vorrichtung und ein Verfahren zur Reduzierung der Taktzeit einer Zentraleinheit (CPU), basierend auf dem Echtzeit-Aktivitätsniveau innerhalb der CPU eines tragbaren Computers.

Beschreibung des Standes der Technik

Die unter der Nummer WO A 86 00432 veröffentlichte internationale Patentanmeldung offenbart einen Computer mit einem veränderlichen Frequenztakt. Der Computer führt eine Tabelle, die die Anzahl der Aufträge, die abzuarbeiten sind, aufnimmt, wobei diese nach der Anzahl an Zyklen, die sie für ihre Ausführung brauchen, in Kategorien unterteilt sind. Wenn Aufträge abgefragt und vollendet sind, wird die Tabelle aktualisiert. Regelmäßig wird eine Zahl für die gewünschte Betriebsfrequenz der CPU aus den Zahlen in der Tabelle und den Gewichtungsfaktoren für jede Kategorie berechnet. Die Frequenz des Taktes wird dann entsprechend eingestellt.

Das IBM Technical Disclosure Bulletin, Vol. 29, Nr. 9, Seiten 4122 bis 4124 offenbart einen Computer mit einer CPU, die einen Ruhezustandsmodus aufweist, in dem die Taktfrequenz minimal ist. Nachdem der Computer länger als eine bestimmte Zeitdauer darauf gewartet hat, daß eine Taste auf der Tastatur durch den Benutzer gedrückt wurde, schaltet er sich selbst in den Ruhezustandsmodus. Auf die volle Verarbeitungsgeschwindigkeit wird auf ein Unterbrechungssignal von der Tastatur hin zurückgestellt, wenn eine Taste gedrückt wird.

Die europäische Patentanmeldung, die unter der Nr. EP 0 363 567 A2 am 18. April 1990 veröffentlicht wurde, offenbart ein Computersystem mit einem veränderlichen Frequenztakt. Auf ein Unterbrechungssignal hin wählt ein Unterbrechungssignalbehandlungsmodul des Betriebssystems eine geeignete Geschwindigkeit für den Takt aus, z.B., um den Prozessor in Synchronisation mit einer Eingabe/Ausgabe-Anpassungskarte zu halten. Wenn das Unterbrechungssignal verarbeitet wurde, wird die vorherige Taktgeschwindigkeit wiederhergestellt.

Im Laufe des Entwicklungsprozesses von Personalcomputern sind die transportablen oder tragbaren Computer sehr populär geworden. Ein solcher tragbarer Computer verwendet eine große Stromversorgung, obwohl er eigentlich einen kleinen Schreibtisch-Personalcomputer darstellt. Tragbare Computer sind kleiner und leichter als ein Schreibtisch-Personalcomputer und ermöglichen es dem Benutzer, die gleiche Software zu verwenden, die auf einem Schreibtisch-Computer verwendet werden kann.

Die erste Generation "tragbarer" Rechner wurde lediglich mit Wechselstrom aus der Steckdose betrieben. Mit fortschreitender Entwicklung der Personalcomputer wurden batteriegespeiste Computer entwickelt. Darüber hinaus wurde eine echte Tragbarkeit durch die Entwicklung einer neuen Anzeigetechnologie, besserer Festplattenspeicher und leichter Komponenten möglich.

Jedoch wurde die entwickelte Software darauf ausgelegt, auf einem Schreibtisch-Personalcomputer zu laufen, mit allen Eigenschaften eines Computers, ohne Rücksicht auf die batteriegespeisten tragbaren Computer, die lediglich über eine begrenzte Energiemenge für kurze Zeitperioden verfügten. Es wurde von der Software, dem Betriebssystem (MS-DOS), dem grundlegenden Eingabe/Ausgabe-System (BIOS) oder der Anwendungssoftware von Dritten keine besondere Rücksicht genommen, um Leistungseinsatz bei diesen tragbaren Computern

einzusparen.

Indem mehr und mehr in hohem Maße funktionale Softwarepakete entwickelt wurden, erfuhren die Schreibtisch-Computerbenutzer ein gesteigertes Leistungsvermögen durch die Einführung von schneller rechnenden CPUs, größerem Speicher und schnelleren Hochleistungs-Festplatten.

Unglücklicherweise liefen die tragbaren Computer weiterhin nur mit Wechselstromversorgung oder mit großen und schweren Batterien. Bei dem Versuch den Leistungsanforderungen der Schreibtisch-Computer und der neuen Software standzuhalten, wurden teure Komponenten verwendet, um den Leistungsanforderungen gerecht zu werden. Trotzdem reichten die schweren Batterien immer noch nicht sehr lang. Das bedeutete für die Benutzer von tragbaren Computern, daß sie sich mit dem Netzbetrieb oder sehr kurzem Batteriebetrieb begnügen mußten, um das Leistungsvermögen zu besitzen, das von der Software Dritter erwartet wurde.

Die Entwickler von tragbaren Computern verminderten das Leistungsvermögen herunter auf das Niveau von Prozessoren vom 8088- und 8086-Typ, um den Leistungsbedarf zu vermindern. Die unterstützenden Schaltungen und die CPU brauchten weniger Leistung zum Laufen, und es konnten daher leichtere Batterien verwendet werden. Unglücklicherweise lief die neue Software, die dem 80286-Typ angepaßte Befehle erforderte, die nicht in den älteren langsameren 8088/8086-CPU's existierten, nicht.

Bei einem Versuch einen tragbaren Computer zu entwickeln, der Leistung einsparen kann, wobei er dadurch einen längeren Batterieeinsatz, kleinere Einheiten und weniger Gewicht bietet, machten sich einige Entwickler von tragbaren Computern daran, den Leistungsbedarf eines tragbaren Rechners zu vermindern, während ein Anwender den Computer nicht benutzt. Z.B. erzielen die Entwickler eine Verminderung des Leistungseinsatzes durch Verlangsamen oder Anhalten der

Festplatte nach einer vorbestimmten Dauer der Inaktivität; wenn die Festplatte nicht verwendet wird, wird die Festplatte abgeschaltet oder einfach in einen Schlummermodus versetzt. Wenn der Anwender bereit ist, die Festplatte einzusetzen, muß der Bediener warten, bis die Festplatte ihre Betriebsgeschwindigkeit erreicht hat und das Computersystem wieder zum Vollbetrieb bereit ist, bevor der Bediener mit der Arbeit fortfahren kann.

Andere Entwickler von tragbaren Computern sparen Leistung ein, indem sie die Computeranzeige ausschalten, wenn die Tastatur nicht verwendet wird. Jedoch setzt der Computer im normalen Betrieb die volle Leistung ein. Mit anderen Worten ist die Leistungseinsparung durch dieses Verfahren lediglich praktikierbar, wenn der Anwender die Komponenten des Systems nicht nutzt. Es ist jedoch sehr wahrscheinlich, daß der Anwender den Computer ausschalten wird, wenn er nicht verwendet wird.

Trotz alledem wird also eine wirkliche Leistungseinsparung benötigt, während der Bediener den Computer für sinnvolles Arbeiten verwendet. Wenn der Bediener den Computer verwendet, ist der Vollbetrieb aller Komponenten erforderlich. In den Intervallen, in denen der Bediener den Computer nicht verwendet, kann der Computer jedoch ausgeschaltet oder verlangsamt werden, um Leistung einzusparen. Für das Aufrechterhalten des Leistungsvermögens ist es entscheidend, zu bestimmen, wann der Computer verlangsamt oder abgeschaltet wird, ohne die Arbeit des Anwenders zu unterbrechen, die Software von Dritten durcheinanderzubringen oder das Betriebssystem zu stören, solange der Betrieb erforderlich ist.

Ferner ist es so, daß obwohl ein Anwender, wie oben beschrieben, warten kann, bis die Festplatte angelaufen ist, die Anwendungssoftwarepakete nicht darauf warten kann, bis die CPU "angelaufen" und bereit ist. Die CPU muß bereit sein, wenn die Anwendungsprogramme Berechnungen durchführen

müssen. Das Umschalten auf Vollbetrieb muß schnell und ohne Beeinträchtigung des Anwendungsprogrammes vollzogen sein. Dieser augenblickliche Übergang muß sowohl für den Anwender als auch für die gegenwärtig aktive Anwendung transparent sein. Verzögerungen verursachen sowohl Bedienungsprobleme für den Anwender in bezug auf die Ansprechzeit und die Softwarekompatibilität, als auch ein generelles Versagen des Computers, ein gefordertes Programm richtig auszuführen.

Zu den weiteren Ansätzen zur Leistungseinsparung bei tragbaren Computern gehört das Vorsehen eines "Herunterfahrens" oder "Bereitsteh-Modus" des Betriebs. Das Problem besteht wieder darin, daß der Computer während dieser Zeit nicht für den Bediener nutzbar ist. Der Bediener könnte genauso gut den Stromversorgungsschalter des Gerätes abschalten, um Leistung zu sparen. Diese Art von Leistungseinsparung erlaubt es lediglich, den tragbaren Computer "herunterzufahren" und dadurch Leistung einzusparen, wenn der Bediener es vergißt, den Stromversorgungsschalter auszuschalten, oder er sich von dem Computer für die programmierte Zeitdauer entfernt. Der Vorteil dieser Art von Leistungseinsparung gegenüber dem bloßen Ausschalten des Stromversorgungsschalters Aus/An ist eine viel schnellere Rückkehr zum Vollbetrieb. Jedoch ist dieses Verfahren der Leistungseinsparung immer noch keine in Echtzeit wirkende intelligente Leistungseinsparung, welche, während der Computer in Betrieb ist und Daten verarbeitet, das Betriebssystem, das BIOS und irgendein Anwenderprogramm von Dritten nicht stört, das gegenwärtig auf dem Computer läuft.

Einige Versuche, um diesem Erfordernis zu begegnen, wurden von VLSI-Lieferanten gemacht, indem sie Schaltungen lieferten, die entweder die Taktsignale zur CPU abschalteten, wenn der Anwender auf der Tastatur keine Eingaben machte, oder den Computer auf Anfrage aufweckten, wenn eine Taste gedrückt wurde. Jeder dieser Ansätze reduzierte die Leistung, jedoch ist der Computer während dieser Zeitdauer tot (nicht verwendbar). Hintergrundoperationen wie das

Aktualisieren der Systemuhr, Übertragungen, Druckwarteschlangen und andere ähnliche Operationen können nicht durchgeführt werden. Einige existierende tragbare Computer verwenden diese Schaltungen. Nach einer programmierten Zeitdauer ohne Aktivität schaltet sich der Computer selbst ab. Der Bediener muß die Anlage erneut einschalten, braucht jedoch nicht das Betriebssystem und das Anwendungsprogramm neu zu starten. Der Vorteil dieser Schaltung liegt - wie bei den bestehenden "Herunterfahr"-Operationen - in einer schnellen Rückkehr zum Vollbetrieb ohne Neustarten des Computers. Nichtsdestotrotz reduziert dieses Verfahren den Leistungsbedarf lediglich dann, wenn sich der Anwender von der Anlage entfernt und verlängert nicht tatsächlich die Betriebsdauer der Batterieladung.

ZUSAMMENFASSUNG DER ERFINDUNG

Angesichts der obigen Probleme, die mit dem Stand der Technik verbunden sind, besteht eine Aufgabe der vorliegenden Erfindung darin, eine Vorrichtung und ein Verfahren zur Echtzeit-Einsparung von Leistung für Computersysteme zu liefern, ohne die Echtzeit-Leistungsfähigkeit zu verringern, wobei eine solche Einsparung von Leistung für den Benutzer transparent bleibt.

Eine weitere Aufgabe der vorliegenden Erfindung besteht darin, eine Vorrichtung und ein Verfahren zum Vorhersagen des Aktivitätsniveaus innerhalb eines Computersystems und zur Verwendung der Vorhersage für eine automatische Leistungseinsparung zu liefern.

Noch eine weitere Aufgabe der vorliegenden Erfindung besteht darin, eine Vorrichtung und ein Verfahren zu liefern, die dem Anwender die Modifikation der automatischen Vorhersagen des Aktivitätsniveaus und die Verwendung der modifizierten Vorhersagen für automatische Leistungseinsparung ermöglichen.

Eine weitere Aufgabe der vorliegenden Erfindung besteht darin, eine Vorrichtung und ein Verfahren zur Echtzeit-Ver-minderung und -Wiederherstellung von Taktgeschwindigkeiten zu liefern, wodurch die CPU auf die volle Verarbeitungs-geschwindigkeit von einer Periode der Inaktivität aus zu-rückgesetzt wird, die für Softwareprogramme transparent ist.

Diese Aufgaben werden bei einer bevorzugten Ausführungsform der vorliegenden Erfindung durch eine Vorrichtung und ein Verfahren erreicht, die, basierend auf dem Aktivi-tätsniveau der CPU, bestimmen, ob eine CPU ruhen kann und, basierend auf dieser Bestimmung, ein Hardware-Auswahlelement aktivieren. Wenn die CPU ruhen oder schlummern kann, legt das Hardware-Auswahlelement Schwingungen bei einem Ruhezus-standstaktniveau an; wenn die CPU aktiv sein soll, wendet das Hardware-Auswahlelement Schwingungen bei einem Hochge-schwindigkeitstaktniveau an.

Gemäß einem ersten Gesichtspunkt der Erfindung wird ein Verfahren zum Implementieren eines Echtzeit-Leistungsein-sparungsmodus in einem Computer mit einer Zentraleinheit (CPU) geliefert, das die folgenden Schritte umfaßt:

Bestimmen des Niveaus der CPU-Aktivität in Echtzeit durch Erkennen von Perioden der Aktivität und Perioden der Inaktivität der CPU;

Entscheiden aufgrund des Niveaus der CPU-Aktivität, ob sie für Leistungseinsparung verfügbar ist;

Veranlassen eines Hardware-Ausfallelementes, die Taktfrequenz unter die aktuelle Taktfrequenz zu reduzieren oder den zur CPU gelieferten Takt zu stoppen, wenn die CPU für Leistungseinsparung verfügbar ist;

Belassen des Taktes im gestoppten Zustand oder bei der reduzierten Frequenz, bis ein Unterbrechungssignal er-scheint, und

Veranlassen des Hardware-Ausfallelementes, die zur CPU gelieferte Taktfrequenz als Reaktion auf das Unterbrechungs-signal auf die aktuelle Taktfrequenz zurückzustellen.

Gemäß einem zweiten Aspekt der Erfindung wird eine Vorrichtung zur Echtzeit-Leistungseinsparung in einem Computer mit einer Zentraleinheit (CPU) geliefert, die umfaßt:

- einen ersten Taktoszillator, um Impulse mit einer ersten Frequenz zu liefern;

- einen zweiten Taktoszillator, um Impulse mit einer zweiten Frequenz zu liefern, die langsamer als die erste Frequenz ist; und

- ein Hardware-Auswahlelement, um Impulse von dem ersten Oszillator oder von dem zweiten Oszillator auszuwählen, wobei das Hardware-Auswahlelement dafür eingerichtet ist, die ausgewählten Impulse als Taktsignale zur CPU zu liefern; wobei die Vorrichtung dadurch gekennzeichnet ist, daß sie darüber hinaus umfaßt:

- ein CPU-Aktivitätserkennungselement zum Bestimmen des Niveaus der CPU-Aktivität in Echtzeit durch Erkennen von Perioden der Aktivität und Perioden der Inaktivität der CPU und Liefern einer Anzeige dieses Niveaus der CPU-Aktivität; und

- ein CPU-Ruhezustandsverwaltungselement, das dafür eingerichtet ist, die Anzeige des Niveaus der CPU-Aktivität von dem CPU-Aktivitätserkennungselement zu empfangen, und ein Signal zum Hardware-Auswahlelement zu liefern, das bestimmt, welche Impulse das Hardware-Auswahlelement auswählen soll, um Taktsignale gemäß der empfangenen Anzeige zur CPU zu liefern.

Die vorliegende Erfindung untersucht sowohl den Zustand der CPU-Aktivität als auch die Aktivität des Bedieners und jedes gegenwärtig aktiven Anwendungssoftwareprogramms. Dieses Abtasten der Aktivität wird in Echtzeit durchgeführt, wobei das Leistungsniveau des Computers eingestellt wird, um die Leistungseinsparung und die Computerleistung zu verwalten. Diese Einstellungen werden innerhalb der CPU-Zyklen erzielt und beeinflussen nicht die Wahrnehmung des Anwenders von der Leistungsfähigkeit.

Daher wird, wenn der Bediener der Software Dritter für das Betriebssystem/BIOS den Computer nicht verwendet, die

vorliegende Erfindung ein schnelles Ausschalten oder Verlangsamens der CPU, solange erforderlich, bewirken, wodurch der Leistungseinsatz vermindert wird, und sie wird sofort, wenn erforderlich, den vollen CPU-Betrieb wiederherstellen, ohne das wahrgenommene Leistungsvermögen zu beeinträchtigen. Dieses Zurückschalten in den Vollbetrieb von dem "Verlangsamungs-" Modus aus tritt auf, ohne daß der Benutzer es anfordern muß, und ohne irgendeine Verzögerung des Betriebs des Computers, bei der darauf gewartet werden muß, daß der Computer zu einem "Bereit"-Zustand zurückzukehrt.

Diese und andere Merkmale und Vorteile der Erfindung werden für Fachleute aus der folgenden detaillierten Beschreibung einer bevorzugten Ausführungsform zusammen mit den begleitenden Zeichnungen offensichtlich, in denen:

BESCHREIBUNG DER ZEICHNUNGEN:

Fig. 1 ein Flußdiagramm ist, das den Selbsteinstellungsgesichtspunkt einer bevorzugten Ausführungsform der vorliegenden Erfindung darstellt;

Fig. 2a-2d Flußdiagramme sind, die das aktive Leistungseinsparungsüberwachungselement beschreiben, das bei der vorliegenden Erfindung eingesetzt wird;

Fig. 3 ein vereinfachtes schematisches Schaubild ist, das die mit der aktiven Leistungseinsparung verbundene Hardware darstellt, die bei der vorliegenden Erfindung verwendet wird;

Fig. 4 ein Schema der Ruhezustands-Hardware für eine Ausführungsform der vorliegenden Erfindung ist; und

Fig. 5 ein Schema der Ruhezustands-Hardware für eine weitere Ausführungsform der vorliegenden Erfindung ist.

GENAUE BESCHREIBUNG EINER BEVORZUGTEN AUSFÜHRUNGSFORM

Untersucht man die Dauer der Computeraktivität in einem bestimmten System, so besitzen die CPU und die zugehörigen Komponenten einen Nutzungsprozentanteil. Wenn der Anwender Daten von der Tastatur eingibt, ist die Zeit zwischen den Tastenanschlägen, gemessen in CPU-Zyklen, sehr lang. Viele

Dinge können während dieser Zeit durch den Computer erfüllt werden, z.B. das Drucken eines Berichtes. Selbst während des Druckens eines Berichtes ist noch Zeit für zusätzliche Operationen wie das im Hintergrund ablaufende Aktualisieren einer Uhr/Kalender-Anzeige verfügbar. Dennoch gibt es fast immer übrige Zeit, wenn die CPU nicht verwendet wird. Wenn der Computer während dieser übrigen Zeit ausgeschaltet oder verlangsamt wird, dann wird die Leistungseinsparung in Echtzeit erhalten. Solche Echtzeit-Leistungseinsparung verlängert die Batterielebensdauer.

Gemäß der bevorzugten Ausführungsform der vorliegenden Erfindung wird eine Kombination aus Hardware und Software benötigt, um Leistung unter MS-DOS oder anderen Betriebssystemen wie OS/2, XENIX und denen für Apple Computer einzusparen. Da die vorliegende Erfindung in jedem System arbeiten wird, ist es zu erkennen daß, obwohl die Implementierung von System zu System leicht variieren kann, der Schutzzumfang der vorliegenden Erfindung daher nicht auf Computersysteme begrenzt werden soll, die unter MS-DOS arbeiten.

Verlangsamen oder Anhalten der Komponenten des Computersystems gemäß der bevorzugten Ausführungsform der vorliegenden Erfindung vermindert den Leistungsbedarf, obwohl die Menge an eingesparter Leistung variieren kann. Gemäß der vorliegenden Erfindung vermindert daher das Anhalten des Taktes (wo dieses möglich ist, da bei einigen CPUs die Taktsignale nicht gestoppt werden können) den Leistungsbedarf mehr als das bloße Verlangsamen des Taktes.

Im allgemeinen kann die Anzahl der Operationen (oder Befehle) pro Sekunde ungefähr als proportional zum Prozessortakt angesehen werden:

$$\text{Befehle/Sekunde} = \text{Befehle/Zyklus} \times \text{Zyklen/Sekunde}$$

Wird zur Vereinfachung angenommen, daß der gleiche Befehl wiederholt ausgeführt wird, so daß der Ausdruck Befehle/Se-

kunde konstant ist, so kann die Beziehung folgendermaßen ausgedrückt werden:

$$F_q = K_1 \times \text{Clk},$$

wobei F_q den Befehlen pro Sekunde entspricht, K_1 eine Konstante ist, die den Befehlen pro Zyklus entspricht, und Clk den Zyklen pro Sekunde entspricht. Daher wächst grob gesprochen die Ausführungsgeschwindigkeit mit der Frequenz des CPU-Taktes.

Die Menge an Leistung, die zu einem bestimmten Moment verwendet wird, steht ebenfalls mit der Frequenz des CPU-Taktes und daher mit der Ausführungsgeschwindigkeit in Beziehung. Im allgemeinen kann diese Beziehung folgendermaßen ausgedrückt werden:

$$P = K_2 + (K_3 \times \text{Clk}),$$

wobei P die Leistung in Watt ist, K_2 eine Konstante in Watt ist, K_3 eine Konstante ist, die die Anzahl der Watt-Sekunden pro Zyklus ausdrückt, und Clk den Zyklen pro Sekunde des CPU-Taktes entspricht. Daher kann auch gesagt werden, daß mit dem Anstieg der CPU-Taktfrequenz die Leistung wächst, für die zu einem bestimmten Zeitpunkt Bedarf besteht.

Angenommen, daß eine bestimmte Zeitperiode T so in N -Intervalle eingeteilt wird, daß die Leistung P in jedem Intervall konstant war. Dann wäre die während T verbrauchte Energiemenge E gegeben durch:

$$E = P(1)\delta T_1 + P(2)\delta T_2 \dots + P(N)\delta T_N$$

Ferner werde angenommen, daß der CPU-Takt " Clk " nur zwei Zustände aufweist, entweder " AN " oder " AUS ". Für die Zwecke dieser Erläuterung stelle der " AN "-Zustand den CPU-Takt bei seiner maximalen Frequenz dar, während der " AUS "-Zustand die minimale Taktgeschwindigkeit darstellt, bei der

die CPU arbeiten kann (diese kann bei CPUs, deren Taktsignale gestoppt werden können, Null sein). In dem Fall, in dem der CPU-Takt immer "AN" ist, sind alle $P(i)$ in der vorhergehenden Gleichung gleich und die Gesamtenergie ist:

$$\begin{aligned} E(\max) &= P(\text{an}) \times (\Delta T_1 + \Delta T_2 \dots + \Delta T_N) \\ &= P(\text{an}) \times T \end{aligned}$$

Das stellt den maximalen Leistungsbedarf des Computers dar, wenn keine Leistungseinsparungsschritte eingesetzt werden. Wenn der CPU-Takt während eines Teils der Intervalle "aus" ist, dann sind zwei Leistungsniveaus für jedes Intervall möglich. $P(\text{an})$ stellt die Leistung dar, die benötigt wird, wenn sich der Takt in seinem "AN"-Zustand befindet, während $P(\text{aus})$ die Leistung darstellt, die eingesetzt wird, wenn der Takt "AUS" ist. Wenn alle Zeitintervalle, in denen der Takt "AN" ist, zu dem Wert " $T(\text{an})$ " aufsummiert werden und die "AUS"-Intervalle zu " $T(\text{aus})$ " aufsummiert werden, dann folgt:

$$T = T(\text{an}) + T(\text{aus})$$

Nun kann man für die Energie, die während der Periode T verwendet wird, schreiben:

$$E = [P(\text{an}) \times T(\text{an})] + [P(\text{aus}) \times T(\text{aus})]$$

Unter diesen Bedingungen kann die verbrauchte Gesamtenergie durch Vergrößern der Zeitintervalle $T(\text{aus})$ vermindert werden. Somit kann durch Steuern der Zeitperioden, in denen sich der Takt in seinem "AUS"-Zustand befindet, die eingesetzte Energiemenge vermindert werden. Wenn die $T(\text{aus})$ -Perioden in eine große Anzahl von Intervallen während der Periode T aufgeteilt werden, dann ist der Energieverbrauch, da die Breite jedes Intervalls gegen Null geht, maximal. Umgekehrt nimmt die verbrauchte Energie ab, wenn die Breite der $T(\text{aus})$ -Intervalle wächst.

Wenn die "AUS"-Intervalle so angeordnet werden, daß sie mit den Perioden, in denen die CPU normalerweise inaktiv ist, zusammenfallen, dann kann der Anwender keine Verminderung der Leistungsfähigkeit wahrnehmen, und der Gesamtenergieverbrauch wird ausgehend von seinem E(max)-Zustand vermindert. Um die T(aus)-Intervalle auf die Perioden der CPU-Inaktivität auszurichten, wird das CPU-Aktivitätsniveau verwendet, um die Breite der T(aus)-Intervalle in einem geschlossenen Regelkreis zu bestimmen. Die Fig. 1 stellt einen solchen geschlossenen Regelkreis dar. Das Aktivitätsniveau der CPU wird beim Schritt 10 bestimmt. Wenn dieses Niveau gegenüber der unmittelbar vorhergehenden Bestimmung angewachsen ist, vermindert die vorliegende Erfindung das T(aus)-Intervall (Schritt 20) und kehrt wieder zur Bestimmung des Aktivitätsniveaus der CPU zurück. Wenn andererseits das Aktivitätsniveau gegenüber einer unmittelbar vorhergehenden Bestimmung vermindert wurde, vergrößert die vorliegende Erfindung das T(aus)-Intervall (Schritt 30) und fährt wieder damit fort, das Aktivitätsniveau der CPU zu bestimmen. Daher werden die T(aus)-Intervalle dauernd eingestellt, um sich an das Aktivitätsniveau des Systems anzupassen.

Bei jedem Betriebssystem gibt es zwei wesentliche Logikpunkte: Eine LEERLAUF- oder "Nichtstun"-Schleife innerhalb des Betriebssystems und einen Anforderungskanal des Betriebssystems, normalerweise verfügbar für Versorgung, die von der Anwendungssoftware benötigt wird. Indem die Logik in Reihe zu diesen Logikpunkten gesetzt wird, kann der Typ der Aktivitätsanforderung, die von einer Anwendungssoftware gestellt wird, ausgewertet werden, die Leistungseinsparung aktiviert werden und Streifendauern bestimmt werden. Eine Streifendauer ist die Anzahl von T(an)- gegenüber T(aus)-Intervallen über die Zeit, berechnet durch das Aktivitätsniveau. Eine Annahme kann gemacht werden, um das CPU-Aktivitätsniveau zu bestimmen: Softwareprogramme, die Versorgung benötigen, benötigen normalerweise zusätzliche Versorgungseingaben, und die Zeitdauer zwischen Versorgungsanforderungen kann verwendet werden, um das Aktivitätsniveau irgend-

einer Anwendungssoftware zu bestimmen, die auf dem Computer läuft, und um die Anzahl von Streifen für die Leistungseinsparung gemäß der vorliegenden Erfindung zu liefern.

Ist die CPU während eines Leistungseinsparungsstreifens (T(aus)) unterbrochen worden, sichert sie den Zustand der unterbrochenen Routine vor dem Zeigen auf die Unterbrechungssoftware. Da die Leistungseinsparungssoftware während dieses Streifens arbeitete, wird die Steuerung natürlich zu der aktiven Leistungseinsparungsschleife (Überwacher 40) zurückgegeben, die einfach den Takt der CPU überwacht, um eine Abbruchbedingung für den Leistungseinsparungsmodus zu bestimmen, wodurch vom T(aus)- zum T(an)-Zustand übergegangen wird. Das Intervall des nächsten Leistungseinsparungszustands wird, wie oben in Verbindung mit Fig. 1 besprochen, durch das überwachte Aktivitätsniveau eingestellt. Einige Implementierungen können einen automatischen Abbruch von T(aus) durch die Hardwarelogik erzeugen, wodurch ein automatischer Ausstieg aus der Leistungseinsparungsschleife erzwungen wird und ein Intervall T(an) ausgeführt wird.

Noch deutlicher wird das, wenn nun die Fig. 2a-2d betrachtet werden, die den aktiven Leistungseinsparungsüberwacher 40 der vorliegenden Erfindung darstellen. Die CPU installiert den Überwacher 40 entweder über ein Programm, das in der CPU-ROM gespeichert ist, oder lädt ihn von einem externen Gerät, indem sie das Programm im RAM speichert. Nachdem die CPU den Überwacher 40 geladen hat, fährt er bei INIT 50 mit einer Systemunterbrechungsinitialisierung, einer anwenderkonfigurationsmäßigen Einrichtung und einer system-/anwendungsspezifischen Initialisierung fort. Der LEERLAUF-Zweig 60 (noch genauer in Fig. 2b dargestellt) wird durch ein Hardware- oder Softwareunterbrechungssignal für eine Leerlauf- oder "Nichtstun"-Funktion ausgeführt. Diese Art von Unterbrechung wird durch die CPU verursacht, die entweder in eine Leerlauf- oder eine "Nichtstun"-Schleife (d.h. geplante Inaktivität) eintritt. Der AKTIVITÄT-Zweig 70 des Flußdiagramms, der noch vollständiger unten mit Bezug auf

Fig. 2d beschrieben ist, wird durch eine Software- oder Hardwareunterbrechung durch eine Betriebssystem- oder Eingabe/Ausgabe-Versorgungsanforderung, durch ein Anwendungsprogramm oder eine interne Betriebssystemsfunktion ausgeführt werden. Eine Eingabe/Ausgabe-Versorgungsanforderung, die durch ein Programm gegeben wird, kann z.B. Festplatten-Eingabe/Ausgabe, Lesen, Drucken, Laden, etc. sein. Unabhängig von dem ausgewählten Zweig wird die Steuerung schließlich zum CPU-Betriebssystem bei "ZURÜCK" 80 zurückgegeben. Der INIT-Zweig 50 dieses Flußdiagramms, der in Fig. 2a dargestellt ist, wird lediglich einmal ausgeführt, wenn er mittels eines Programmes in den ROM geladen wird, oder er wird jedesmal während des Einschaltens ausgeführt, wenn er von einem externen Gerät geladen und in den RAM gespeichert wird. Ist dieser Zweig des aktiven Leistungsüberwachers 40 einmal vollständig ausgeführt worden, wenn die Steuerung von dem Betriebssystem zum Leistungseinsparungsmodus übergeben wurde, wird, abhängig von dem Typ der CPU-Aktivität einer der beiden Zweige LEERLAUF 60 oder AKTIVITÄT 70 ausgewählt: Der LEERLAUF-Zweig 60 für Leistungseinsparung während geplanter Inaktivität und der AKTIVITÄT-Zweig 70 für Leistungseinsparung während CPU-Aktivität.

Unter näherer Betrachtung des INIT-Zweiges 50 ist zu erkennen, daß die Routine, nachdem alle Systemunterbrechungen und -variablen initialisiert sind, beim Schritt 90 fortfährt, um das Leistungsniveau auf das Vorgabenniveau einzustellen. Bei Betriebssystemen, wo der Anwender über eine Eingabesteuerung für das Leistungsniveau verfügt, führt das Programm beim Schritt 100 eine Überprüfung durch, um zu sehen, ob ein Anwenderniveau ausgewählt wurde. Wenn das Anwenderniveau kleiner als Null oder größer als das maximale Niveau ist, verwendet das System das Vorgabenniveau. Im anderen Falle fährt es beim Schritt 110 fort, wobei es das Leistungsniveau so modifiziert, daß es dem Anwenderniveau entspricht.

Gemäß der bevorzugten Ausführungsform der vorliegenden

Erfindung stellt das System beim Schritt 120 die Variable LEERLAUF_TICK auf Null und die Variable AKTIVITÄT_TICK auf Null. Bei einer MS/DOS-Implementierung bezieht sich LEERLAUF_TICK auf die Anzahl der Unterbrechungen, die in einer "Nichtstun"-Schleife gefunden wurden. AKTIVITÄT_TICK bezieht sich auf die Anzahl der Unterbrechungen, die durch eine Aktivitätsunterbrechung verursacht wurden, was wiederum das CPU-Aktivitätsniveau bestimmt. Die Tickanzahl stellt einen Zeit-Deltawert für die nächste Unterbrechung dar. LEERLAUF_TICK ist ein konstanter Zeit-Deltawert von einem Ticken (Unterbrechung) zu einem anderen dar, wenn nicht durch eine Softwareunterbrechung überschrieben. Eine Softwareunterbrechung kann den Zeit-Deltawert zwischen Unterbrechungen neu programmieren.

Nach dem Einstellen der Variablen auf Null fährt die Routine bei der Einrichtung 130 fort, bei der irgendeine anwendungsspezifische Konfigurationsfeineinstellung hinsichtlich systemspezifischer Einzelheiten behandelt wird und das System initialisiert wird. Als nächstes beaufschlagt die Routine die Unterbrechung Eingabe/Ausgabe (Schritt 140) mit Befehlen für die Hardware, die anzeigen, daß die Hardware die Steuerung bei der nächsten Unterbrechung übernehmen kann. Der INIT-Zweig 50 geht dann vom Betriebssystem oder was auch immer den aktiven Leistungsüberwacher ursprünglich aufgerufen hatte, beim Schritt "ZURÜCK" 80 weg.

Man betrachte nun den LEERLAUF-Zweig 60 des aktiven Leistungsüberwachers 40, der noch ausführlicher in Fig. 2b beschrieben ist. Als Reaktion auf eine geplante Inaktivität der CPU führt der Überwacher 40 (in dieser Figur nicht speziell dargestellt) eine Überprüfung durch, um zu sehen, ob der Zugang zum LEERLAUF-Zweig 60 erlaubt ist, indem zuerst bestimmt wird, ob die Aktivitätsunterbrechung gegenwärtig beschäftigt ist. Wenn BESCHÄFTIGT_A dem BESCHÄFTIGT_FLAG entspricht (Schritt 150), welches ein Rücksprungflag ist, dann ist die CPU beschäftigt und kann nicht in den Ruhezustand versetzt werden. Daher geht der Überwacher 40 sofort

zu "ZURÜCK I" 160 über und verläßt die Routine. "ZURÜCK I" 160 ist ein indirekter Vektor auf die vorhergehende LEERLAUF-Vektorunterbrechung des Betriebssystems für die normale Verarbeitung, gespeichert vor dem Erreichen des Überwachers 40 (d.h. dieses verursacht einen Unterbrechungsrücksprung zum letzten verketteten Vektor).

Wenn das BESCHÄFTIGT_A-Unterbrechungsflag nicht gesetzt ist, dann führt der Überwacher 40 eine Überprüfung durch, um zu sehen, ob das BESCHÄFTIGT_LEERLAUF-Unterbrechungsflag, BESCHÄFTIGT_I, dem BESCHÄFTIGT_FLAG entspricht (Schritt 170). Wenn das so ist, so zeigt das an, daß sich das System schon im LEERLAUF-Zweig 60 des Überwachers 40 befindet, und daher sollte sich das System nicht selbst unterbrechen. Wenn BESCHÄFTIGT_I = BESCHÄFTIGT_FLAG gilt, dann verläßt das System die Routine am indirekten Vektor "ZURÜCK I" 160.

Wenn jedoch weder das BESCHÄFTIGT_A-Rücksprungflag noch das BESCHÄFTIGT_I-Rücksprungflag gesetzt wurden, setzt die Routine beim Schritt 180 das BESCHÄFTIGT_I-FLAG zum Rücksprungschutz (BESCHÄFTIGT_I = BESCHÄFTIGT_FLAG). Beim Schritt 190 wird LEERLAUF_TICK um eins erhöht. LEERLAUF_TICK ist die Anzahl der T(an)-Intervalle vor einem T(aus)-Intervall und wird aus LEERLAUF-Unterbrechungen, Einrichtungs-Unterbrechungen und von dem Aktivitätsniveau der CPU bestimmt. LEERLAUF_TICK wird um eins erhöht, um eine Glättung von Ereignissen zu erlauben, wodurch eine kritische Eingabe/Ausgabe-Aktivitätssteuerungsglättung ermöglicht wird.

Beim Schritt 200 führt der Überwacher 40 eine Überprüfung durch, um zu sehen, ob LEERLAUF_TICK LEERLAUF_MAXTICKS entspricht. LEERLAUF_MAXTICKS ist eine der Konstanten, die bei der Einrichtung 130 des INIT-Zweigs 50 initialisiert wurden, bleibt für ein System konstant und ist verantwortlich für die Selbsteinstellung des Aktivitätsniveaus. Wenn LEERLAUF_TICK nicht LEERLAUF_MAXTICKS entspricht, dann wird das BESCHÄFTIGT_I-FLAG beim Schritt 210 gelöscht und es wird die Schleife verlassen, indem zum indirekten Vektor "ZURÜCK

I" 160 übergegangen wird. Wenn jedoch LEERLAUF_TICK LEERLAUF_MAXTICKS entspricht, dann wird LEERLAUF_TICK auf LEERLAUF_START_TICKS eingestellt (Schritt 220). LEERLAUF_START_TICKS ist eine Konstante, die Null entsprechen kann oder Null nicht entsprechen kann (je nachdem, ob bei der speziellen CPU der Takt gestoppt werden kann). Dieser Schritt bestimmt die Selbsteinstellung wie oft der Rest der Ruhezustandsfunktionen durchgeführt werden kann. Durch Einstellen von LEERLAUF_START_TICKS auf LEERLAUF_MAXTICKS minus 1 wird ein kontinuierliches T(aus)-Intervall erzielt. Beim Schritt 230 wird das Leistungsniveau überprüft. Wenn es gleich Null ist, dann löscht der Überwacher das BESCHÄFTIGT_I-Flag (Schritt 210), verläßt die Routine beim "ZURÜCK I" 160 und gibt die Steuerung zum Betriebssystem zurück, so daß es mit dem fortfahren kann, was es ursprünglich ausgeführt hat, bevor es zu dem aktiven Leistungsüberwacher 40 übergegangen ist.

Wenn jedoch das Leistungsniveau beim Schritt 240 nicht gleich Null ist, dann bestimmt die Routine, ob eine Unterbrechungsmaske am Platz ist. Eine Unterbrechungsmaske wird durch die System/Anwendungssoftware eingestellt und bestimmt, ob Unterbrechungen für den Überwacher 40 verfügbar sind. Wenn Unterbrechungen NICHT_VERFÜGBAR sind, dann wird das BESCHÄFTIGT_I-Rücksprungflag gelöscht und die Steuerung wird zum Betriebssystem zurückgegeben, damit es mit dem fortfahren kann, was es ausgeführt hat, bevor es zum Überwacher 40 übergegangen ist. Sowohl die Betriebssysteme als auch die Anwendungssoftware können das T(an)-Intervall einstellen, so daß ein kontinuierlicher T(an)-Zustand geliefert wird, indem die Unterbrechungsmaske auf NICHT_VERFÜGBAR eingestellt wird.

Unter der Annahme, daß ein Unterbrechungssignal VERFÜGBAR ist, fährt der Überwacher 40 mit der Leistungseinsparungsunterroutine 250 fort, die vollständig während einer T(aus)-Periode ausgeführt wird, die durch den Hardwarezustand errichtet wird. (Z.B. bei der bevorzugten Ausführungs-

form der vorliegenden Erfindung kann das längstmögliche Intervall 18 ms betragen, welches die längste Zeitdauer zwischen zwei Ticks oder Unterbrechungen von dem Echtzeit-Takt darstellt.) Während der Leistungseinsparungsunterroutine 250 wird der CPU-Takt auf ein Ruhezustandstaktniveau heruntergesetzt.

Wenn eine kritische Eingabe/Ausgabe-Operation die T(an)-Intervalle erzwingt, ist das Unterbrechungssignal des LEERLAUF-Zweigs 60 darauf gerichtet, für zusätzliche kritische Eingabe/Ausgabe-Anforderungen bereit zu bleiben. Wenn die CPU mit kritischen Eingaben/Ausgaben beschäftigt wird, sind weniger T(aus)-Intervalle verfügbar. Umgekehrt sind, wenn kritische Eingabe/Ausgabe-Anforderungen abnehmen und die Zeitintervalle zwischen ihnen anwachsen, mehr T(aus)-Intervalle verfügbar. Der LEERLAUF-Zweig 60 ist ein selbsteinstellendes System, das auf der Rückmeldung von den Aktivitätsunterbrechungssignalen basiert, und er ist darauf gerichtet, mehr T(aus)-Intervalle zu liefern, wenn sich das Aktivitätsniveau verlangsamt. Sobald der Überwacher 40 die Leistungseinsparungsunterroutine 250 abgeschlossen hat, was in Fig. 2c und noch ausführlicher unten beschrieben wird, wird das BESCHÄFTIGT_I-Rücksprungflag gelöscht (Schritt 210) und die Steuerung wird beim Schritt "ZURÜCK I" 160 zu dem jeweiligen Betriebssystem zurückgegeben, das ursprünglich den Überwacher 40 anforderte.

Man betrachte nun Fig. 2c, die ein Flußdiagramm ist, das die Leistungseinsparungsunterroutine 250 darstellt. Der Überwacher 40 bestimmt, welches der Eingabe/Ausgabe-Hardware-Hochgeschwindigkeitstakt beim Schritt 260 ist. Er stellt die AKTUELLE_TAKT_FREQUENZ auf den relevanten Hochgeschwindigkeitstakt ein und sichert diesen Wert zur Verwendung für CPUs mit Mehrfachniveau-Hochgeschwindigkeitstaktgebern. Wenn daher eine bestimmte CPU Hochgeschwindigkeitstaktgeber mit 12 MHz und 6 MHz besitzt, muß der Überwacher 40 entscheiden, auf welchem Hochgeschwindigkeitstakt sich die CPU befindet, bevor der Überwacher 40 die

Leistung reduziert, so daß die CPU auf den richtigen Hochgeschwindigkeitstakt zurückgesetzt werden kann, wenn die CPU aufwacht. Beim Schritt 270 wird die EINSPARUNGS_TAKT_FREQUENZ auf die bestimmte AKTUELLE_TAKT_FREQUENZ eingestellt. Die EINSPARUNGS_TAKT_FREQUENZ 270 wird nicht verwendet, wenn es nur einen Hochgeschwindigkeitstaktgeber für die CPU gibt. Der Überwacher 40 fährt nun beim Ruhezustandstakt 280 fort, wobei ein Impuls zum Hardware-Auswahlelement (in Fig. 3 dargestellt) gesendet wird, um den CPU-Takt in den Ruhezustand zu versetzen (d.h. seine Taktfrequenz zu verlangsamen oder anzuhalten). Der Eingabe/Ausgabe-Port-Hardware-Ruhezustandstakt liegt bei wesentlich weniger Schwingungen als der normalerweise verwendete CPU-Takt.

An diesem Punkt kann eines von zwei Ereignissen eintreten. Eine System/Anwendungsunterbrechung kann auftreten oder eine Echtzeit-Taktunterbrechung kann auftreten. Wenn eine System/Anwendungsunterbrechung 290 auftritt, fährt der Überwacher 40 mit der Unterbrechungsroutine 300 fort, verarbeitet die Unterbrechung sobald wie möglich, beaufschlagt die Unterbrechung Eingabe/Ausgabe beim Schritt 310 und kehrt zurück, um zu bestimmen, ob es eine Unterbrechung gegeben hat (Schritt 320). Da es in diesem Fall eine Unterbrechung gegeben hat, wird die EINSPARUNGS_TAKT_FREQUENZ verwendet (Schritt 330), um zu bestimmen, auf welchen Hochgeschwindigkeitstakt die CPU zurückgestellt wird, und die Leistungseinsparungsunterroutine 250 wird beim "ZURÜCK" 340 verlassen. Wenn jedoch ein System/Anwendungsunterbrechungssignal nicht empfangen wurde, fährt die Leistungseinsparungsunterroutine 250 damit fort, zu warten, bis eine Echtzeit-Taktunterbrechung aufgetreten ist (Schritt 320). Wenn eine solche Unterbrechung aufgetreten ist, setzt die Leistungseinsparungsunterroutine 250 die CPU wieder auf die gespeicherte EINSPARUNGS_TAKT_FREQUENZ zurück. Wenn die Ruhezustandstaktfrequenz nicht angehalten wurde, mit anderen Worten, wenn die Ruhezustandstaktfrequenz nicht Null war, dann wird die Steuerung bei einer niedrigen Frequenz übergeben, und die Leistungseinsparungsunterroutine 250 wird die

Unterbrechungsschleife 320 mehrere Male ausführen. Falls jedoch die Steuerung übergeben wurde, als die Ruhezustandstaktfrequenz Null war, mit anderen Worten, wenn es keinen Takt gab, wird die Leistungseinsparungsunterroutine 250 die Unterbrechungsschleife 320 einmal ausführen, bevor sie den CPU-Takt auf die EINSPARUNGS_TAKT_FREQUENZ 330 zurückstellt und aussteigt (Schritt 340).

Man betrachte nun die Fig. 2d, die ein Flußdiagramm darstellt, das den AKTIVITÄT-Zweig 70 zeigt, der durch eine Anwendungs/System-Aktivitätsanforderung über eine Betriebssystem-Versorgungsanforderungsunterbrechung ausgelöst wird. Der AKTIVITÄT-Zweig 70 beginnt mit dem Rücksprungschutz. Der Überwacher 40 bestimmt beim Schritt 350, ob BESCHÄFTIGT_I auf BESCHÄFTIGT_FLAG eingestellt wurde. Wenn das so ist, dann bedeutet das, daß sich das System schon im LEERLAUF-Zweig 60 befindet und nicht unterbrochen werden kann. Wenn BESCHÄFTIGT_I = BESCHÄFTIGT_FLAG ist, dann geht der Überwacher 40 zu "ZURÜCK I" 160 weiter, welches ein indirekter Vektor auf eine Vektorunterbrechung einer alten Aktivität für Normalverarbeitung ist, über einen Unterbrechungsvektor, nachdem das Betriebssystem die angeforderte Versorgung liefert.

Wenn jedoch das BESCHÄFTIGT_I-Flag nicht dem BESCHÄFTIGT_FLAG entspricht, was bedeutet, daß der LEERLAUF-Zweig 60 nicht erreicht wird, bestimmt der Überwacher 40 beim Schritt 360, ob das BESCHÄFTIGT_A-Flag gleich dem BESCHÄFTIGT_FLAG gesetzt wurde. Wenn das so ist, dann wird die Steuerung an diesem Punkt zum System zurückgegeben, da der AKTIVITÄT-Zweig 70 schon verwendet wird und nicht unterbrochen werden kann. Wenn das BESCHÄFTIGT_A-Flag nicht gesetzt wurde, mit anderen Worten BESCHÄFTIGT_A nicht BESCHÄFTIGT_FLAG entspricht, setzt der Überwacher 40 beim Schritt 370 BESCHÄFTIGT_A = BESCHÄFTIGT_FLAG, so daß er während der Ausführung des AKTIVITÄT-Zweiges 70 nicht unterbrochen wird. Beim Schritt 380 wird das Leistungsniveau bestimmt. Wenn das Leistungsniveau Null ist, steigt der

Überwacher 40 aus dem AKTIVITÄT-Zweig 70 aus, nachdem er das BESCHÄFTIGT_A-Rücksprungflag gelöscht hat (Schritt 390). Wenn jedoch das Leistungsniveau nicht gleich Null ist, dann wird als nächstes die AKTUELLE_TAKT_FREQUENZ der Eingabe/Ausgabe-Hardware bestimmt. Analog zum Schritt 270 der Fig. 2c wird beim Schritt 400 der Fig. 2d die AKTUELLE_TAKT_FREQUENZ verwendet, wenn es Hochgeschwindigkeitstakte bei mehreren Niveaus für eine gegebene CPU gibt. Ist das nicht der Fall, so entspricht die AKTUELLE_TAKT_FREQUENZ immer dem CPU-Hochgeschwindigkeitstakt. Nachdem die AKTUELLE_TAKT_FREQUENZ bestimmt wurde (Schritt 400), wird beim Schritt 410 LEERLAUF_TICK auf die Konstante START_TICKS eingestellt, die für die vorher bestimmte AKTUELLE_TAKT_FREQUENZ eingerichtet wurde. T(aus)-Intervalle werden, basierend auf dem aktuellen aktiven Hochgeschwindigkeitstakt, eingerichtet.

Der Überwacher 40 bestimmt als nächstes, ob eine Anforderung gemacht wurde. Eine Anforderung ist eine Eingabe durch die Anwendungssoftware, die auf dem Computer läuft, für einen bestimmten Typ von benötigter Versorgung. Beim Schritt 420 bestimmt der Überwacher 40, ob die Anforderung eine kritische Eingabe/Ausgabe ist. Wenn die Anforderung eine kritische Eingabe/Ausgabe ist, wird sie T(an) in kontinuierlicher Weise dazu zwingen, verlängert zu werden, bis T(an) größer als T(aus) ist, und der Überwacher 40 wird aus dem AKTIVITÄT-Zweig 70 aussteigen, nachdem er das BESCHÄFTIGT_A-Rücksprungflag gelöscht hat (Schritt 390). Wenn die Anforderung anderenfalls keine kritische Eingabe/Ausgabe ist, dann wird der AKTIVITÄT_TICK beim Schritt 430 um eins erhöht. Es wird dann beim Schritt 440 bestimmt, ob der AKTIVITÄT_TICK nun AKTIVITÄT_MAXTICKS entspricht. Der Schritt 440 erlaubt eine Glättung von einer kritischen Eingabe/Ausgabe und macht das System bereit ausgehend von einer weiteren kritischen Eingabe/Ausgabe während AKTIVITÄT_TICK-T(an)-Intervallen. Unter der Annahme, daß AKTIVITÄT_TICK nicht AKTIVITÄT_MAXTICKS gleicht, wird der AKTIVITÄT-Zweig 70 nach dem Löschen des BESCHÄFTIGT_A-Rücksprungflags verlassen.

(Schritt 390). Wenn andererseits AKTIVITÄT_TICK der Konstante AKTIVITÄT_MAXTICKS gleicht, wird beim Schritt 450 AKTIVITÄT_TICK auf die Konstante NIVEAU_MAXTICKS eingestellt, die für das spezielle, beim Schritt 380 bestimmte Leistungsniveau errichtet wurde.

Nun bestimmt der Überwacher 40, ob eine Unterbrechungs-
maske existiert (Schritt 460). Eine Unterbrechungs-
maske wird durch die System/Anwendungs-Software eingestellt. Wird sie
auf NICHT_VERFÜGBAR eingestellt, so wird ein kontinuierli-
cher T(an)-Zustand erzeugt. Wenn die Unterbrechungs-
maske NICHT_VERFÜGBAR gleicht, gibt es keine verfügbaren Unter-
brechungen zu diesem Zeitpunkt, und der Überwacher 40 steigt
aus dem AKTIVITÄT-Zweig 70 nach dem Löschen des BESCHÄF-
TIGT_A-Rücksprungflags aus (Schritt 390). Wenn jedoch eine
Unterbrechung VERFÜGBAR ist, bestimmt der Überwacher 40 beim
Schritt 470, ob die Anforderung, die beim Schritt 420 er-
kannt wurde für eine langsame Eingabe/Ausgabe_UNTERBRECHUNG
bestimmt war. Langsame Eingabe/Ausgabe-Anforderungen können
eine Verzögerung aufweisen, bis das Eingabe/Ausgabe-Gerät
"bereit" wird. Während dieser Operation des "Bereitmachens"
kann ein kontinuierliches T(aus)-Intervall eingestellt und
ausgeführt werden, um Leistung einzusparen. Daher wird, wenn
die Anforderung keine LANGSAME Eingabe/Ausgabe_UNTERBRECHUNG
ist, der AKTIVITÄT-Zweig 70 nach dem Löschen des BESCHÄF-
TIGT_A-Rücksprungflags verlassen (Schritt 390). Wenn jedoch
die Anforderung eine LANGSAME Eingabe/Ausgabe_UNTERBRECHUNG
ist und noch Zeit existiert, bevor das Eingabe/Ausgabe-Gerät
"bereit" wird, dann bestimmt der Überwacher 40 beim Schritt
480, ob die Eingabe/Ausgabe-Anforderung vollständig ist
(d.h.: ist das Eingabe/Ausgabe-Gerät bereit?). Wenn das Ein-
gabe/Ausgabe-Gerät nicht bereit ist, zwingt der Überwacher
40 T(aus) dazu, verlängert zu werden, wodurch die CPU ge-
zwungen wird zu warten oder zu schlummern, bis das langsame
Eingabe/Ausgabe-Gerät bereit ist. An diesem Punkt gibt es
Zeit, um Leistung einzusparen, und der AKTIVITÄT-Zweig 70
tritt in die Leistungseinsparungsunterroutine 250 ein, die
vorher in Verbindung mit Fig. 2c beschrieben wurde. Wenn

jedoch die Eingabe/Ausgabe-Anforderung vollständig ist, dann wird die Steuerung zum Betriebssystem zurückgegeben, nachdem der Überwacher 40 aus dem AKTIVITÄT-Zweig 70 nach dem Löschen des BESCHÄFTIGT_A-Rücksprungflags ausgestiegen ist (Schritt 390).

Die Selbsteinstellung ist in dem Steuersystem der kontinuierlichen Rückführungsschleifen inbegriffen. Die Software der vorliegenden Erfindung kann erkennen, wann die CPU-Aktivität gering ist und daher, wann der Leistungseinsparungsaspekt der vorliegenden Erfindung aktiviert werden kann. Wenn der Leistungseinsparungsüberwacher aktiviert ist, wird eine prompte Rückkehr zur vollen Geschwindigkeit der CPU-Taktoperation innerhalb des Intervalls so erzielt, daß die Leistungsfähigkeit des Computers nicht vermindert wird. Um diese prompte Rückkehr zur vollen Geschwindigkeit der CPU-Taktoperation zu erzielen, verwendet die bevorzugte Ausführungsform der vorliegenden Erfindung einiges an zugehöriger Hardware.

Nun wird Fig. 3 betrachtet, die eine vereinfachte schematische Darstellung zeigt, die die zugehörige Hardware darstellt, die bei der vorliegenden Erfindung für aktive Leistungseinsparung verwendet wird. Wenn der (nicht dargestellte) Überwacher 40 feststellt, daß die CPU zum Schlummern bereit ist, schreibt er es in einen (nicht dargestellten) Eingabe/Ausgabe-Port ein, der einen Impuls auf der Ruhezustandsleitung verursacht. Die ansteigende Flanke dieses Impulses auf der Ruhezustandsleitung führt dazu, daß das Flip-Flop 500 einen H-Wert an Q und einen L-Wert an Q- flankengesteuert registriert. Das führt dazu, daß die UND/ODER-Logik (UND-Gatter 510, 520; ODER-Gatter 530) die Impulse von dem Ruhezustandstaktoszillator 540 auswählt, die die Ruhezustandstaktleitung entlanglaufen, damit sie zum CPU-Takt laufen und von diesem verwendet werden. Der Ruhezustandstaktoszillator 540 ist ein langsamerer Taktgeber als der CPU-Taktgeber, der während der normalen CPU-Aktivität verwendet wird. Der H-Wert, der von Q des Flip-Flops 500 kommt UND

(510) die Impulse, die von dem Ruhezustandsoszillator 540 kommen, ODER (530) die Ergebnisse des L-Wertes an dem Q- des Flip-Flops 500 UND (520) die Impulse, die entlang der Hochgeschwindigkeitstaktleitung durch den Hochgeschwindigkeitstaktoszillator 550 erzeugt werden, liefern den CPU-Takt. Wenn der Eingabe/Ausgabe-Port den Ruhezustandstakt bestimmt, dann ist der CPU-TAKT gleich dem Ruhezustandstaktoszillator (540)-Wert. Wenn andererseits eine Unterbrechung auftritt, löscht ein Unterbrechungswert das Flip-Flop 500, wodurch das UND/ODER-Auswahlelement (umfassend 510, 520 und 530) gezwungen wird, den Hochgeschwindigkeitstaktwert auszuwählen, und stellt den CPU-Taktwert auf den Wert zurück, der von dem Hochgeschwindigkeitstaktoszillator 550 kommt. Daher wird während jeder Leistungseinsparungsoperation an der CPU die Erkennung irgendeiner Unterbrechung innerhalb des Systems die CPU-Operation vor dem Zeigen auf die Unterbrechung und der Verarbeitung der Unterbrechung auf die volle Taktgeschwindigkeit zurückstellen.

Es ist zu erkennen, daß die zugehörige benötigte Hardware, die außerhalb jeder der CPUs für ein gegebenes System liegt, basierend auf dem verwendeten Betriebssystem verschieden sein kann, je nachdem, ob die CPU gestoppt werden kann, etc. Nichtsdestoweniger soll der Schutzzumfang der vorliegenden Erfindung nicht durch mögliche systemspezifische Modifikationen begrenzt werden, die erforderlich sind, um es der vorliegenden Erfindung zu ermöglichen, aktiv die Leistung in den zahlreichen verfügbaren tragbaren Computersystemen einzusparen. Als Beispiel sind zwei tatsächliche Implementierungen in Fig. 4 und 5 dargestellt, die unten erläutert werden.

Viele VSLI-Entwürfe erlauben heute das Taktschalten der CPU-Geschwindigkeit. Die Logik, um von einer Null-Takt- oder einer langsamen Takt-Logik zu einer schnellen Takt-Logik umzuschalten ist die gleiche, die es dem Anwender ermöglicht, Geschwindigkeiten durch einen Tastaturbefehl zu verändern. Die zu dem Überwacher 40 hinzugefügte Logik, die mit

einer solchen Schaltlogik arbeitet, verursacht eine sofortige Rückkehr zu einem schnellen Takt nach der Erkennung irgendeiner Unterbrechung. Diese einfache Logik ist der Schlüssel zu der notwendigen Hardwareunterstützung, um die CPU zu unterbrechen und dadurch die Verarbeitung der Unterbrechung bei voller Geschwindigkeit zu ermöglichen.

Das Verfahren, um den Leistungsbedarf unter MS-DOS zu vermindern, verwendet die MS-DOS-LEERLAUF-Schleifen-Falle, um Zugang zu der "Nichtstun"-Schleife zu erhalten. Die LEERLAUF-Schleife liefert einen speziellen Zugang zu Anwendungssoftware- und Betriebssystemoperationen, die sich in einem Zustand der LEERLAUF-Aktivität oder niedriger Aktivität befinden. Eine sorgfältige Untersuchung ist erforderlich, um das Aktivitätsniveau an irgendeinem bestimmten Punkt innerhalb des Systems zu bestimmen. Rückführungsschleifen werden verwendet von der Unterbrechung-21H-Servicenanforderung, um das Aktivitätsniveau zu bestimmen. Die Vorhersage des Aktivitätsniveaus wird durch Unterbrechung-21H-Anforderungen bestimmt, woraus die vorliegende Erfindung die Streifenperioden zum "Schlafen" (Verlangsamen oder Anhalten) der CPU einstellt. Ein zusätzliches Merkmal erlaubt es dem Anwender, die Streifen je nach dem Aktivitätsniveau der Unterbrechung 21H zu modifizieren.

Nun betrachte man Fig. 4, die ein Schema einer tatsächlichen Ruhezustands-Hardwareimplementierung für ein System wie den Intel 80386 darstellt (der Takt der CPU kann nicht angehalten werden). Der Adreßfreigabebus 600 und der Adreßbus 610 liefern CPU-Eingaben zum Demultiplexer 620. Das Ausgangssignal des Demultiplexers 620 wird entlang SLEEPCS geschickt und als Eingangssignal an die ODER-Gatter 630, 640 geliefert. Die anderen Eingangssignale zu den ODER-Gattern 630, 640 sind die Eingabe/Ausgabe-Schreibsteuerleitung bzw. die Eingabe/Ausgabe-Lesesteuerleitung. Die Ausgangssignale dieser Gatter werden, zusätzlich zum NOR-Gatter 650, an D-Flip-Flop 660 angelegt, um den Port zu dekodieren. "INTR" ist das Unterbrechungseingabesignal von dem Eingabe/Ausgabe-

Port (Peripheriegeräte) in das NOR-Gatter 650, das die Logik-Hardware dazu veranlaßt, zu dem Hochgeschwindigkeitstakt zurückzuschalten. Das Ausgangssignal des Flip-Flops 660 wird dann, zusammen mit dem Ausgangssignal von dem ODER-Gatter 630 dem Tristate-Puffer 670 zugeführt, um ihn freizugeben, um in Rückwärtsrichtung zu lesen, was an dem Port ist. Die gesamte oben gekennzeichnete Hardware wird von dem Lesen/Schreiben-Eingabe/Ausgabe-Port (Peripheriegeräte) verwendet, um die Leistungseinsparungs-"Ruhezustands"-Operation auszuwählen. Das Ausgangssignal "SLOW-" entspricht dem "SLEEP" in Fig. 2 und wird in das Flip-Flop 680 wie später diskutiert, eingegeben.

Das Ausgangssignal des Ruhezustandstaktoszillators 690 wird durch die D-Flip-Flops 700, 710 in zwei langsamere Takte aufgeteilt. Bei der speziellen Implementierung, die in Fig. 4 gezeigt ist, wird der 16 MHz-Ruhezustandstaktoszillator 690 in 4-MHz- und 8-MHz-Taktgeber eingeteilt. Der Jumper J1 wählt aus, welcher Takt der "Ruhezustandstakt" sein soll.

Bei dieser speziellen Implementierung ist der Hochgeschwindigkeitstaktoszillator 720 ein 32 MHz Oszillator, obwohl diese spezielle Geschwindigkeit kein Erfordernis der vorliegenden Erfindung ist. Der 32 MHz Oszillator wird in Reihe mit einem Widerstand (bei der dargestellten Implementierung 33 Ohm) geschaltet, der in Reihe mit zwei parallelen Kondensatoren (10 pF) liegt. Das Ergebnis dieser Oszillationen wird mit den Taktflanken der D-Flip-Flops 730, 740 verbunden.

Die D-Flip-Flops 680, 730 und 740 sind synchronisierende Flip-Flops; 680 und 730 waren in der vereinfachten Ruhezustandshardware der Fig. 2 nicht dargestellt. Diese Flip-Flops werden verwendet, um sicherzustellen, daß die Taktschaltung lediglich bei Taktkanten auftritt. Wie in Fig. 4 zu erkennen ist, wie beim Flip-Flop 500 der Fig. 2, aktiviert das Ausgangssignal des Flip-Flops 740 entweder das

ODER-Gatter 750 oder das ODER-Gatter 760, abhängig davon, ob die CPU schlafen ("FASTEN-") oder aufwachen ("SLOWEN-") soll.

Die ODER-Gatter 750, 760 und das UND-Gatter 770 sind die funktionalen Äquivalente zu dem UND/ODER-Auswahlelement der Fig. 2. Sie sind zuständig für das Auswählen entweder der "slowclk" (langsamer Takt, auch bekannt als Ruhezustandstakt) oder des Hochgeschwindigkeitstaktes (bezeichnet mit 32 MHz auf der ankommenden Leitung). Bei dieser Implementierung beträgt der langsame Takt entweder 4 MHz oder 8 MHz, abhängig von dem Jumper J1, und der Hochgeschwindigkeitstakt beträgt 32 MHz. Das Ausgangssignal des UND-Gatters 770 (ATUCLK) errichtet die Frequenz des CPU-Taktes und entspricht dem CPU-Takt der Fig. 2.

Nun betrachte man Fig. 5, die ein Schema einer weiteren tatsächlichen Ruhezustandshardware-Implementierung für ein System wie den Intel 80286 (der Takt der CPU kann gestoppt werden) darstellt. Der Western Digital FE3600 VLSI wird für die Geschwindigkeitsschaltung mit einem speziellen externen PAL 780 verwendet, um die Unterbrechungstorsteuerung zu steuern, die die CPU bei einer Unterbrechung aufweckt. Die Softwareleistungseinsparung gemäß der vorliegenden Erfindung überwacht die Unterbrechungsakzeptanz, wobei das nächste $P(i)\Delta T_i$ -Intervall nach der Unterbrechung aktiviert wird.

Jede Unterbrechungsanforderung zur CPU wird das System in den Normalbetrieb zurücksetzen. Eine Unterbrechungsanforderung ("INTRQ") zur CPU wird PAL veranlassen, ein Aufwecksignal auf der RESCPU-Leitung zum FE3001 (nicht dargestellt) herauszugeben, das wiederum die CPU und die DMA-Taktgeber freigibt, um das System in seinen normalen Zustand zurückzubringen. Das entspricht der "Unterbrechung" der Fig. 2. Die Unterbrechungsanforderung ist synchronisiert, um zu vermeiden, daß der Zustand der Anlage durcheinandergerät, so daß eine Unterbrechung (INTDET) lediglich erkannt werden wird, während der Zyklus aktiv ist. Die ansteigende Flanke

des RESCPU wird FE 3001 aufwecken, was wiederum das gesamte System aus dem Ruhezustandsmodus befreit.

Die Implementierung für den 386SX ist lediglich in bezug auf die externe Hardware- und Software-Leistungseinsparungsschleife verschieden. Die Softwareschleife wird die äußere Hardware einstellen, so daß sie zu dem Hochgeschwindigkeitstakt bei einer Unterbrechung vor dem Zeigen auf die Unterbrechung schaltet. Wenn zu der Leistungseinsparungssoftware zurückgesprungen wird, wird der Hochgeschwindigkeitstaktzyklus erkannt werden und die Hardware wird auf den vollen Taktbetrieb zurückgestellt werden.

Die Implementierung für OS/2 verwendet die "Nichtstun"-Schleife, die als ein THREAD-Ablauf im Hintergrundbetrieb mit niedriger Priorität programmiert wird. Wenn der THREAD aktiviert ist, wird die CPU-Ruhezustands- oder Niedriggeschwindigkeitstakt-Operation aktiviert werden, bis eine Unterbrechung auftritt, wodurch die CPU auf ihre ursprüngliche Taktgeschwindigkeit zurückgesetzt wird.

Obwohl Unterbrechungssignale verwendet wurden, um die CPU bei der bevorzugten Ausführungsform der vorliegenden Erfindung aufzuwecken, ist zu erkennen, daß jede periodische Aktivität, die sich innerhalb des Systems befindet oder an das System angelegt ist, ebenfalls für die gleiche Funktion verwendet werden könnte.

690 27 510.2-08

EP 0 426 410 (90311832.1-2215)

Patentansprüche

1. Verfahren zum Implementieren eines Echtzeit-Leistungseinsparungsmodus in einem Computer mit einer Zentraleinheit (CPU), das die aufeinanderfolgenden Schritte umfaßt:

Bestimmen des Niveaus der CPU-Aktivität in Echtzeit durch Erkennen von Perioden der Aktivität (70) und Perioden der Inaktivität (60) der CPU;

Entscheiden aufgrund des Niveaus der CPU-Aktivität, ob sie für Leistungseinsparung verfügbar ist;

Veranlassen eines Hardware-Auswahlelementes (500, 510, 520, 530), die Taktfrequenz unter die aktuelle Taktfrequenz zu reduzieren oder den zur CPU (280) gelieferten Takt zu stoppen, wenn die CPU für Leistungseinsparung verfügbar ist;

Belassen des Taktes im gestoppten Zustand oder bei der reduzierten Frequenz bis ein Unterbrechungssignal (320) erscheint, und

Veranlassen des Hardware-Auswahlelementes, die zur CPU gelieferte Taktfrequenz als Reaktion auf das Unterbrechungssignal (330) auf die aktuelle Taktfrequenz zurückzustellen.

2. Verfahren nach Anspruch 1, bei dem der Schritt des Veranlassens eines Hardware-Auswahlelementes, die zur CPU gelieferte Taktfrequenz unter die aktuelle Taktfrequenz zu reduzieren oder das zur CPU (280) gelieferte Taktsignal zu stoppen, darüber hinaus den Schritt umfaßt:

Geben eines CPU-Leistungseinsparungsbefehls über eine Kommunikationsleitung zum Hardware-Auswahlelement;

wobei das Hardware-Auswahlelement dafür eingerichtet ist, einen Leistungseinsparungstaktgeber als Reaktion auf den CPU-Leistungseinsparungsbefehl auszuwählen und Impulse von dem Leistungseinsparungstaktgeber zur CPU zu liefern, wodurch die CPU in einen Leistungseinsparungsmodus gesetzt wird.

3. Verfahren nach Anspruch 1 oder 2, bei dem der Schritt des Veranlassens des Hardware-Auswahlelementes, die zur CPU gelieferte Taktfrequenz auf die aktuelle Taktfrequenz zurückzustellen, darüber hinaus den Schritt umfaßt:

Geben eines CPU-Aktivierungsbefehls über eine Kommunikationsleitung zum Hardware-Auswahlelement;

wobei das Hardware-Auswahlelement veranlaßt wird, den Taktgeber mit der aktuellen Taktfrequenz als Reaktion auf den CPU-Aktivierungsbefehl auszuwählen und Taktimpulse von dem Taktgeber mit der aktuellen Taktfrequenz zur CPU zu liefern, um dadurch die CPU zu aktivieren.

4. Verfahren nach Anspruch 1, Anspruch 2 oder Anspruch 3, bei dem der Schritt des Bestimmens des Niveaus der CPU-Aktivität darüber hinaus die Schritte umfaßt:

Feststellen, ob sich die CPU noch in einem Leistungseinsparungsmodus befindet;

und falls das nicht der Fall ist, Bestimmen, ob Unterbrechungssignale verfügbar sind, um die zur CPU gelieferte Taktfrequenz auf die aktuelle Taktfrequenz zurückzustellen bevor die CPU in einen Leistungseinsparungsmodus gesetzt wird.

5. Verfahren nach Anspruch 1, bei dem der Schritt des Veranlassens eines Hardware-Auswahlelementes, die Taktfrequenz unter die aktuelle Taktfrequenz zu reduzieren oder den zur CPU gelieferten Takt zu stoppen, darüber hinaus die Schritte umfaßt:

Bestimmen, ob es einen Anstieg der CPU-Aktivität gegeben hat; und

Einstellen eines Ruheintervalls der CPU gemäß dieser Bestimmung (20, 30).

6. Verfahren nach Anspruch 1, bei dem das Unterbrechungssignal durch eine periodische Aktivität in dem Computer erzeugt wird.

7. Verfahren nach einem der Ansprüche 1 bis 6, das das Auswählen der aktuellen Taktfrequenz aus entweder einer ersten Taktschwindigkeit, die durch einen ersten Hochgeschwindigkeits-Taktgeber geliefert wird, oder einer zweiten Taktschwindigkeit, die durch einen zweiten Hochgeschwindigkeits-Taktgeber geliefert wird, beinhaltet.

8. Verfahren nach Anspruch 7, bei dem der Schritt des Entschheidens, ob die CPU für Leistungseinsparung verfügbar ist, darüber hinaus die Schritte umfaßt:

Bestimmen der aktuellen Taktfrequenz (260) der CPU, wenn die CPU für Leistungseinsparung verfügbar ist; und

Speichern eines Wertes, der die bestimmte aktuelle Taktfrequenz (270) repräsentiert.

9. Verfahren nach Anspruch 8, bei dem der Schritt des Veranlassens des Hardware-Auswahlelementes, die bestimmte aktuelle Taktfrequenz für die CPU zurückzustellen, darüber hinaus die Schritte umfaßt:

Wiederauffinden des gespeicherten Wertes; und

Zurückstellen der Taktfrequenz der CPU auf die durch den wiederaufgefundenen Wert (330) repräsentierte Frequenz.

10. Verfahren nach einem der Ansprüche 1 bis 9, bei dem vor dem Schritt des Bestimmens des Niveaus der CPU-Aktivität die Steuerung von dem Betriebssystem der CPU entgegengenommen wird und nach dem Schritt des Veranlassens des Hardware-Auswahlelementes, die zur CPU gelieferte Taktfrequenz auf die aktuelle Taktfrequenz zurückzustellen, die Steuerung zum Betriebssystem zurückgegeben wird.

11. Verfahren nach Anspruch 10, bei dem in dem Fall, daß die aus dem Bestimmen des Niveaus der CPU-Aktivität resultierende Entscheidung so ausfällt, daß die CPU für Leistungseinsparung nicht verfügbar ist, die Steuerung dann zum Betriebssystem zurückkehrt.

12. Vorrichtung zur Echtzeit-Leistungseinsparung in einem Computer mit einer Zentraleinheit (CPU), umfassend:

einen ersten Taktoszillator (550), um Impulse mit einer ersten Frequenz zu liefern;

einen zweiten Taktoszillator (540), um Impulse mit einer zweiten Frequenz zu liefern, die langsamer als die erste Frequenz ist; und

ein Hardware-Auswahlelement (500, 510, 520, 530), um Impulse von dem ersten Oszillator oder von dem zweiten Oszillator auszuwählen, wobei das Hardware-Auswahlelement dafür eingerichtet ist, die ausgewählten Impulse als Taktsignale zur CPU zu liefern; wobei die Vorrichtung dadurch gekennzeichnet ist, daß sie darüber hinaus umfaßt:

ein CPU-Aktivitätserkennungselement (10) zum Bestimmen des Niveaus der CPU-Aktivität in Echtzeit durch Erkennen von Perioden der Aktivität und Perioden der Inaktivität der CPU und Liefern einer Anzeige dieses Niveaus der CPU-Aktivität; und

ein CPU-Ruhezustandsverwaltungselement (20, 30), das dafür eingerichtet ist, die Anzeige des Niveaus der CPU-Aktivität von dem CPU-Aktivitätserkennungselement zu empfangen, und ein Signal zum Hardware-Auswahlelement zu liefern, das bestimmt, welche Impulse das Hardware-Auswahlelement auswählen soll, um Taktsignale gemäß der empfangenen Anzeige zur CPU zu liefern.

13. Vorrichtung nach Anspruch 12, bei der das CPU-Aktivitätserkennungselement und das CPU-Ruhezustandsverwaltungselement in der CPU liegen.

14. Vorrichtung nach Anspruch 12 oder Anspruch 13, bei der das CPU-Ruhezustandsverwaltungselement darüber hinaus umfaßt:

ein Einstellungselement (20, 30), das auf die Anzeige reagiert, die das aktuelle erkannte CPU-Aktivitätsniveau repräsentiert, um eine CPU-Ruheperiode zu verlängern, wenn das aktuelle erkannte CPU-Aktivitätsniveau abgenommen hat, oder um die CPU-Ruheperiode zu verkürzen, wenn das aktuelle

erkannte CPU-Aktivitätsniveau zugenommen hat;

ein Taktfrequenzerhaltungselement zum Bestimmen einer aktuellen Taktfrequenz für die CPU, sowohl um einen der aktuellen Taktfrequenz entsprechenden Wert zu speichern, während die CPU Taktpulse mit einer anderen Frequenz zur Leistungseinsparung empfängt, als auch um den gespeicherten Wert wiederaufzufinden, wenn die CPU in den Normalbetrieb zurückkehren soll; und

einen Impulsgenerator zum Erzeugen eines Impulses, um für das Hardware-Auswahlelement zu bestimmen, welche Oszillatorimpulse das Hardware-Auswahlelement auswählen soll.

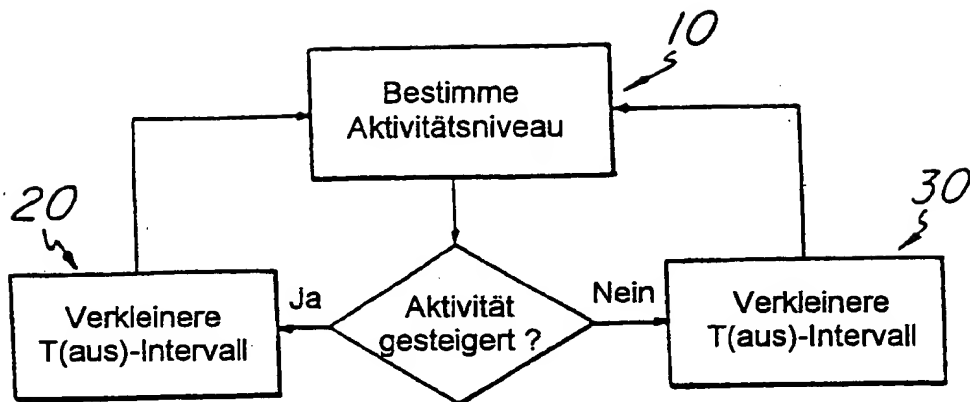


Fig. 1

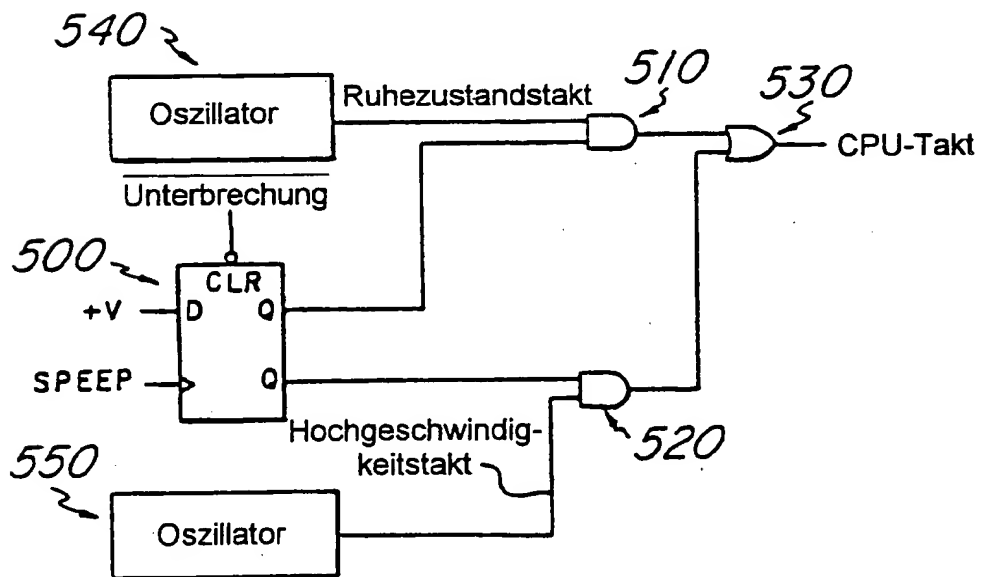


Fig. 3

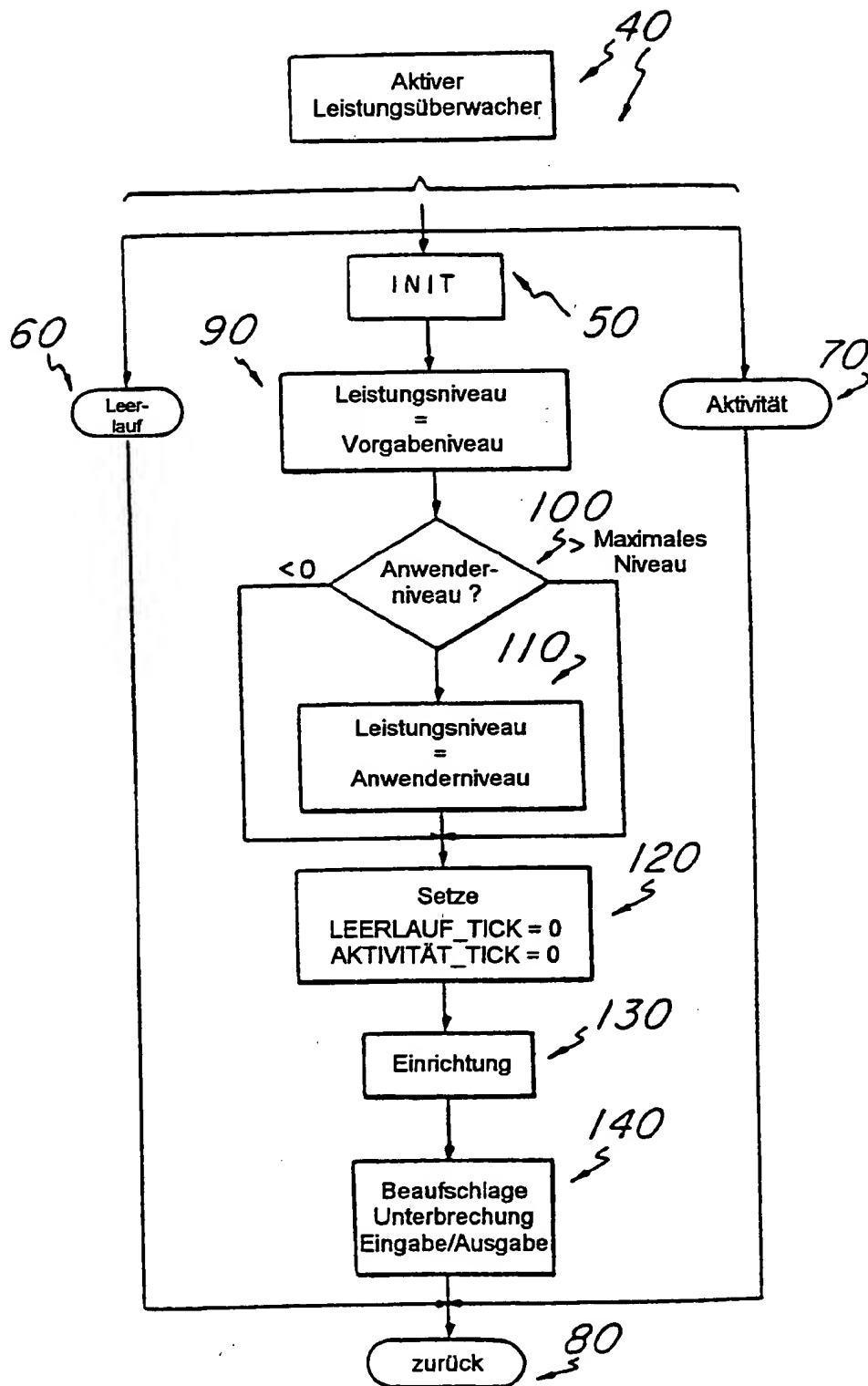


Fig. 2a

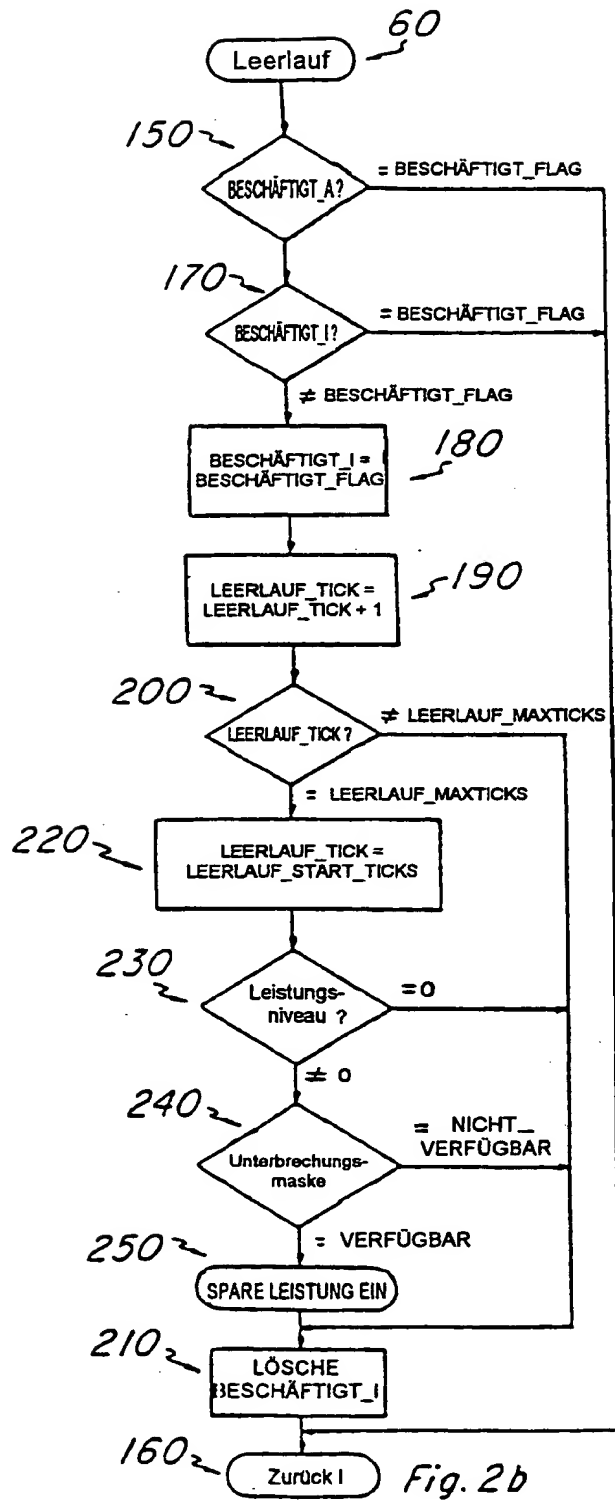


Fig. 2b

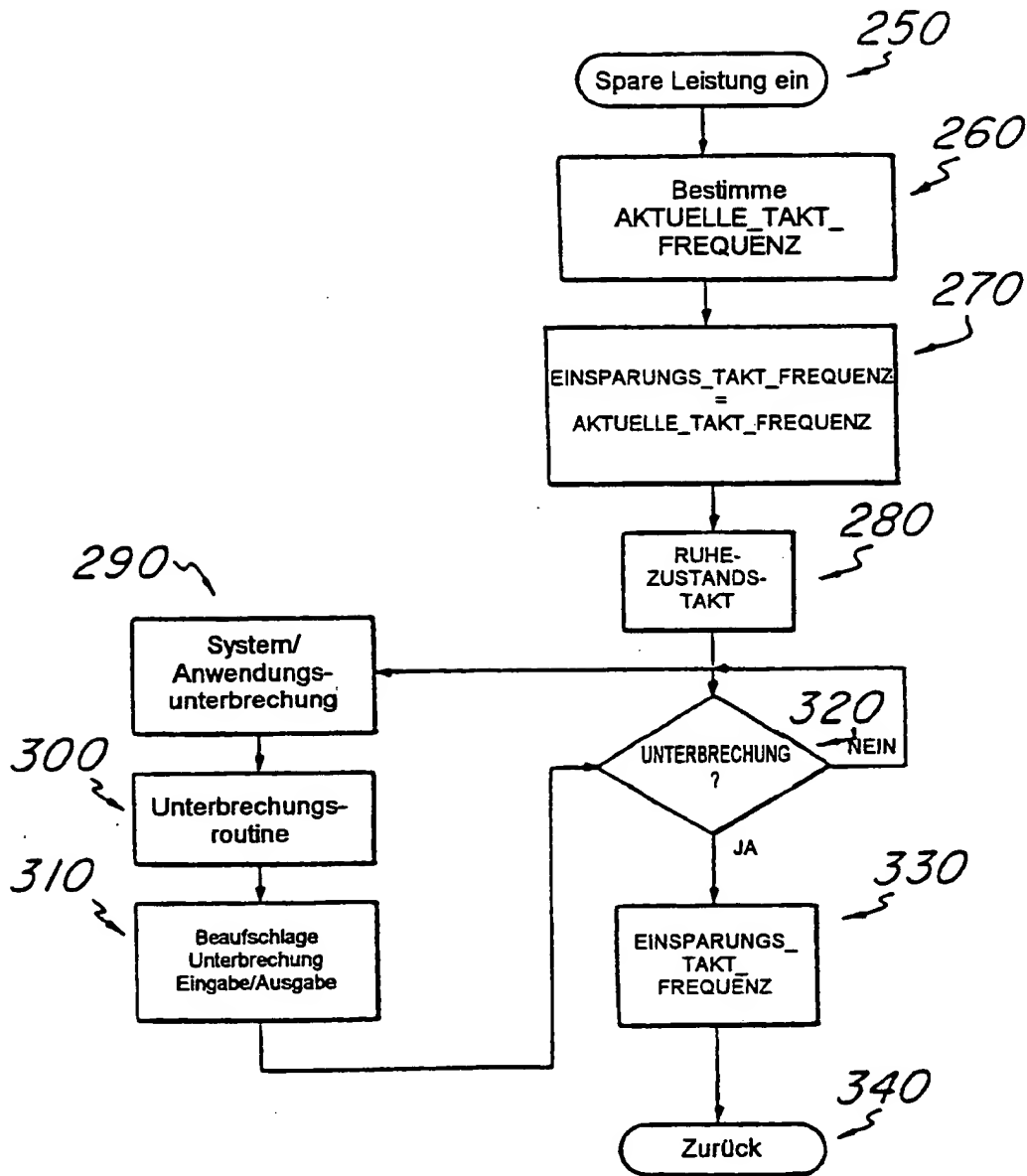


Fig. 2c

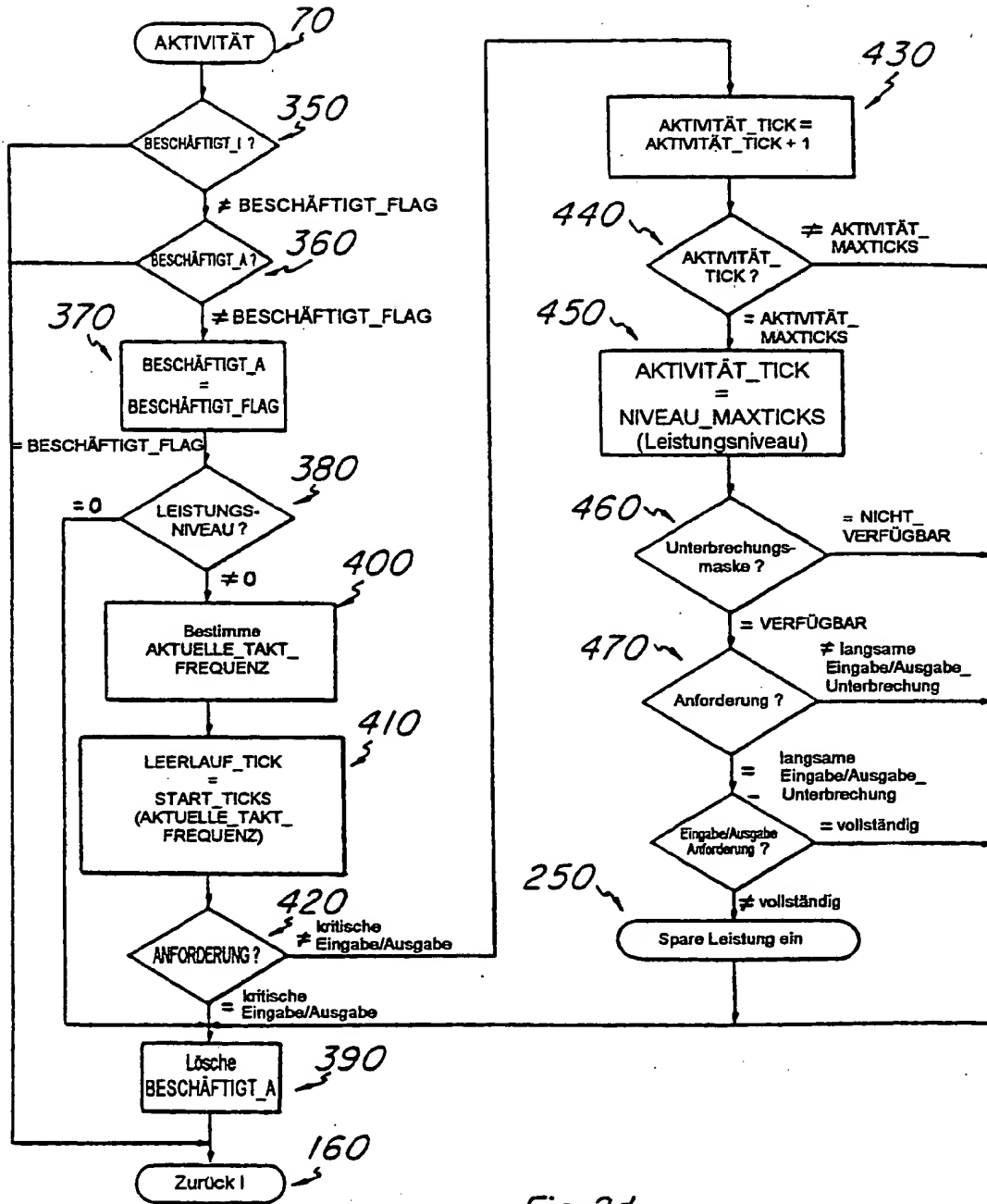


Fig. 2d

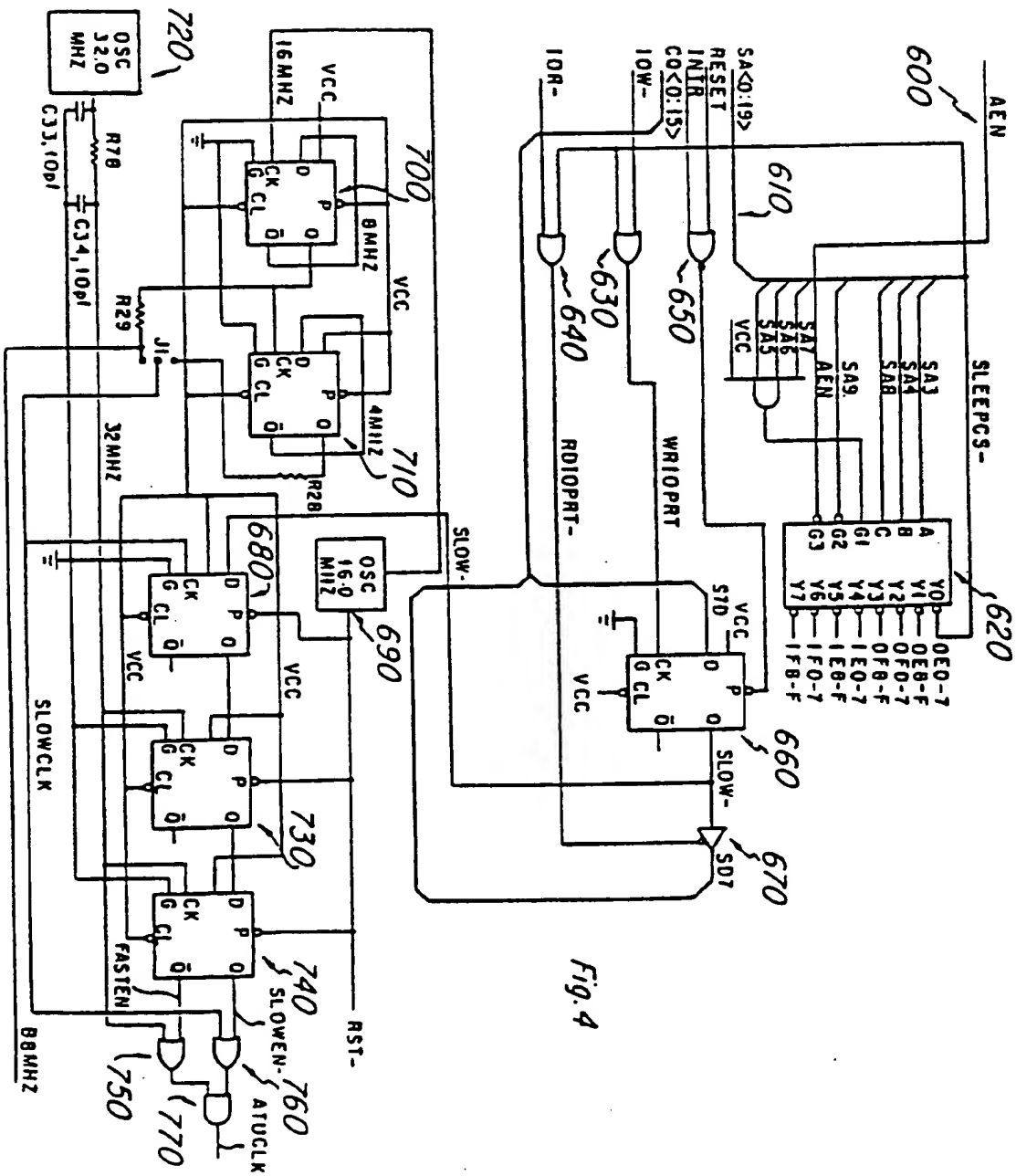


Fig. 4

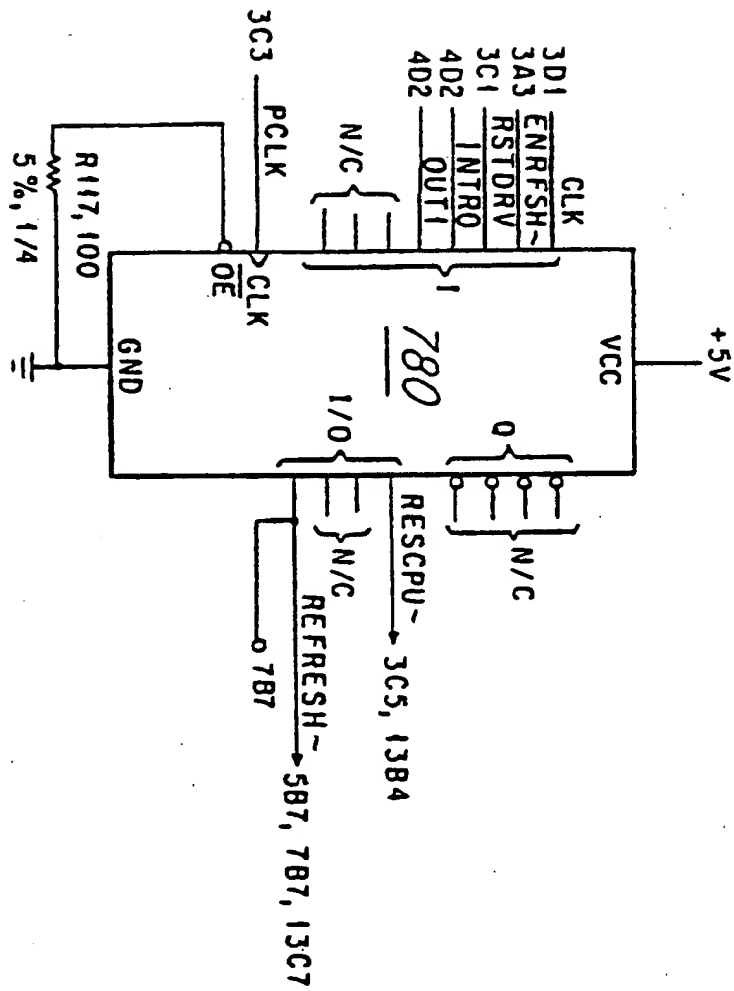


Fig. 5

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

This Page Blank (uspto)